

# Active Versus Passive Transformations in Robotics

BY J.M. SELIG

There are two main ways to keep track of rigid body motions. Most current texts on robotics use the passive approach, where the rigid body has a coordinate frame embedded in it, and then its position and orientation is given by the coordinate transformation from the world frame to a frame moving with the body. When there are several bodies and when bodies also carry several different frames, it can be hard to account for all the different frames.

In the lesser-known active view, there is a single fixed coordinate frame. The position and orientation of a rigid body is specified by the transformation, which moves the body from its home position to its current position. This active view is completely equivalent to the traditional passive view, so readers familiar with the traditional approach need not change their habits. However, newcomers to the subject may find it simpler to learn robotics using the active view since there is a greater emphasis on the bodies and their motion and, consequently, less emphasis on coordinate frames and changes of coordinates.

This article presents the active view and relates it to the traditional passive view. The idea is to show that the active view is straightforward to teach and learn but is entirely equivalent to the standard passive approach.

## Transformation of Points

Let us begin with familiar material, rigid body transformations of points. As mentioned previously, we have a single fixed coordinate frame. Now consider a point  $\mathbf{p}$  in our coordinate system that will have components  $(x, y, z)$  and, hence, will be represented by a column vector

$$\mathbf{p} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}.$$

A rotation around the  $z$ -axis can be modeled by a  $3 \times 3$  matrix:

$$R_z(\theta) = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

where  $\theta$  is the angle of rotation. The effect of such a rotation on point  $\mathbf{p}$  is the matrix product:

$$\mathbf{p}' = R_z(\theta)\mathbf{p}.$$

The coordinates of the new point  $\mathbf{p}'$  will be

$$\mathbf{p}' = \begin{bmatrix} x \cos \theta - y \sin \theta \\ x \sin \theta + y \cos \theta \\ z \end{bmatrix}.$$

Notice that the  $z$ -coordinate is unchanged because we are rotating around the  $z$ -axis.

A translation can be modeled by vector addition. Suppose we want to translate by a translation vector  $\mathbf{t}$ ; the effect on our point  $\mathbf{p}$  will be

$$\mathbf{p}' = \mathbf{p} + \mathbf{t}.$$

In robotics, it is common to combine rotations and translation into  $4 \times 4$  matrices:

$$M = \begin{bmatrix} R & \mathbf{t} \\ 0 & 1 \end{bmatrix}.$$

This is a partitioned matrix with  $R$  a  $3 \times 3$  rotation matrix and  $\mathbf{t}$  a three-dimensional (3-D) column vector corresponding to the translation; note that the 0 in the bottom left corner really represents a row of three zeros. This representation of a rigid transformation is sometimes called the homogeneous representation because of connections with projective geometry, but projective geometry will not be considered here.

To model the effect of such a transformation on a point, we need to change our representation of points. From now on, a point with coordinates  $(x, y, z)$  will be represented by a four-dimensional (4-D) column vector:

$$\tilde{\mathbf{p}} = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{p} \\ 1 \end{bmatrix}.$$

So the effect of a rigid body transformation can be written as the matrix product

$$\tilde{\mathbf{p}}' = M\tilde{\mathbf{p}} = \begin{bmatrix} R\mathbf{p} + \mathbf{t} \\ 1 \end{bmatrix}.$$

So far, everything has been active; the points moved! In the passive view, the points stay still and the coordinate frames move.

To relate the two views, think of a rigid body with a coordinate frame embedded in it. To begin with, that is in the home position, our fixed frame and the frame in the body coincide. After the transformation, the body has a new position and orientation, and, therefore, so does its embedded frame (see Figure 1).

Suppose that  $\mathbf{p}$  is a point in the body. In the home position, it has the coordinates

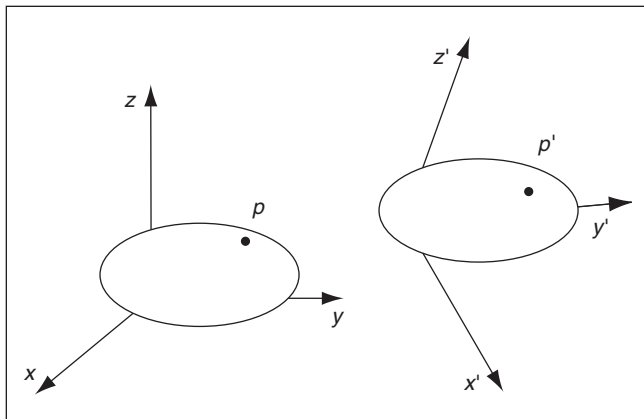


Figure 1. Two coordinate frames.

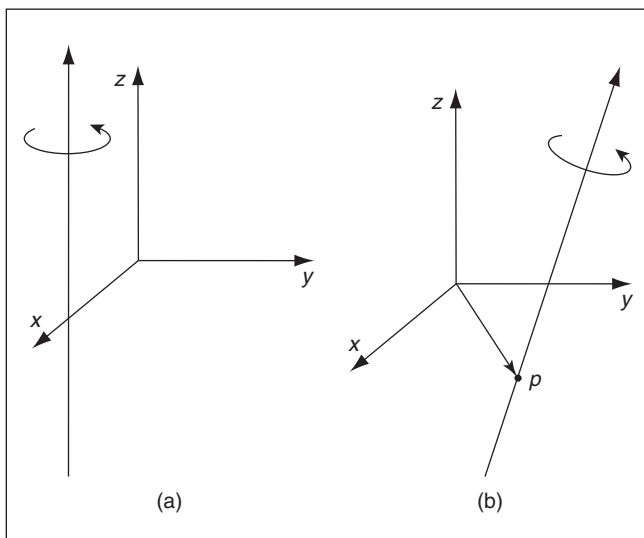


Figure 2. Rotation around two different axes.

$$\mathbf{p} = \begin{bmatrix} x \\ y \\ z \end{bmatrix},$$

relative to the fixed frame and also relative to home position of the body frame since these are the same. Now, when the body moves, the point becomes  $\mathbf{p}'$ , but in the body frame, the coordinates have not changed.

In the passive approach, we need to be able to distinguish between the different coordinate frames. We have to introduce a new layer of notation to express the fact that the vector  $\mathbf{p}$  could be given in coordinates relative to the fixed frame  ${}^f\mathbf{p}$  or the body frame  ${}^b\mathbf{p}$ .

Extending these conventions to the 4-D vectors in the obvious way, we have in the fixed coordinate frame

$${}^f\tilde{\mathbf{p}}' = M {}^f\tilde{\mathbf{p}}.$$

What we want here however, is the coordinate transformation relating coordinates in the two frames. Notice that the point  $\mathbf{p}'$  is in exactly the same position relative to the body frame that  $\mathbf{p}$  was relative to the fixed frame. In coordinates, this means that  ${}^b\mathbf{p}' = {}^f\mathbf{p}$ . Substituting this into the previous equation, we get

$${}^b\tilde{\mathbf{p}}' = M^{-1} {}^f\tilde{\mathbf{p}}'.$$

So the relationship between an active transformation and a passive one is that the coordinate transformation is the inverse of the corresponding active transformation.

Notice that this explanation is very prejudiced towards the active view. The above could also be expressed by saying that the active transformation is the inverse of the corresponding passive transformation.

Finally, notice that the inverse of these partitioned matrices is easy to find. If the original transformation was

$$M = \begin{bmatrix} R & \mathbf{t} \\ 0 & 1 \end{bmatrix},$$

then its inverse will be

$$M^{-1} = \begin{bmatrix} R^T & -R^T\mathbf{t} \\ 0 & 1 \end{bmatrix},$$

where  $R^T = R^{-1}$  is the transpose of the rotation matrix, and this is the inverse of  $R$  since rotational matrices are orthogonal, i.e., they satisfy the relation  $RR^T = I$ .

## Conjugation

Before using this active view to look at the forward kinematics of serial manipulators, it is necessary to look a little more closely at rotations. In general, a rotation in 3-D space is a

rotation around an axis. This rotation axis is a fixed line in space. Let us work out the  $4 \times 4$  matrix that represents a rotation of  $\theta$  radians around a particular line in space. Assume this line is parallel to the  $z$ -axis, and suppose  $(1, 0, 0)$  is a point on the line [see Figure 2(a)]. The rotation around this line is clearly equivalent to the following sequence of rigid transformations. First, we translate the line so that it coincides with the  $z$ -axis, then we rotate around the  $z$ -axis, and, finally, we translate the axis back to its starting position. The advantage of splitting the rotation up in this way is that we already know the matrices representing each of these transformations. The translations are given by matrices of the form

$$T_{\pm} = \begin{bmatrix} 1 & 0 & 0 & \pm 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Taking the minus sign here gives the translation of the line to the  $z$ -axis, it will take the point on the line  $(1, 0, 0)$  to the origin. Taking the plus sign translates the line (and the point on it) back to its original position. The rotation is simply a rotation around the origin

$$R(\theta) = \begin{bmatrix} \cos \theta & -\sin \theta & 0 & 0 \\ \sin \theta & \cos \theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

To combine the three transformations, we multiply the matrices with the first transformation on the right,

$$T_+ R(\theta) T_- = \begin{bmatrix} \cos \theta & -\sin \theta & 0 & 1 - \cos \theta \\ \sin \theta & \cos \theta & 0 & -\sin \theta \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Suppose we want to compute the transformation matrix of a rotation around some general line in space. Assume that  $R$  is the  $3 \times 3$  rotation matrix whose rotation axis is in the direction of the line, and let  $\mathbf{p} = (x, y, z)^T$  be the position vector of some point on the line [see Figure 2(b)]. The  $4 \times 4$  matrix we seek is given in partitioned form by

$$\begin{aligned} A(\theta) &= \begin{bmatrix} I & \mathbf{p} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} R & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} I & -\mathbf{p} \\ 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} R & (I - R)\mathbf{p} \\ 0 & 1 \end{bmatrix}, \end{aligned}$$

where  $I$  represents the  $3 \times 3$  identity matrix. Notice that the point on the line we used is not important, any other point on the line would be given by

$$\mathbf{p}' = \mathbf{p} + \lambda \mathbf{v},$$

where  $\mathbf{v}$  is a vector in the direction of the line and  $\lambda$  is an arbitrary scalar. Since the rotation leaves its axis fixed, we must have  $R\mathbf{v} = \mathbf{v}$  and hence  $(I - R)\mathbf{v} = \mathbf{0}$ . Therefore,

$$(I - R)\mathbf{p}' = (I - R)\mathbf{p}.$$

In group theory, this kind of operation on a transformation is called a conjugation. That is, a conjugation of a transformation  $M$  by a transformation  $N$  is given by the product  $N^{-1}MN$ . Conjugation is a very useful and common operation in group theory.

In robotics, there is another major use of conjugation connected with coordinate transformations. Suppose that  $M$  is an active transformation expressed in a fixed coordinate frame. Assume that the transformation moves the point  ${}^f\tilde{\mathbf{p}}$  to the point  ${}^f\tilde{\mathbf{p}}' = M {}^f\tilde{\mathbf{p}}$ . Now, what is the matrix of this transformation expressed in a different coordinate frame? In this new coordinate frame, the coordinates of the points can be assumed to be  ${}^n\tilde{\mathbf{p}} = N^{-1} {}^f\tilde{\mathbf{p}}$  and  ${}^n\tilde{\mathbf{p}}' = N^{-1} {}^f\tilde{\mathbf{p}}'$ , where  $N$  is the active transformation that moves the fixed frame to the new one. Clearly,

$${}^n\tilde{\mathbf{p}}' = N^{-1} {}^f\tilde{\mathbf{p}}' = N^{-1} M {}^f\tilde{\mathbf{p}} = N^{-1} M N {}^n\tilde{\mathbf{p}},$$

so that the transformation that takes the point  ${}^n\tilde{\mathbf{p}}$  to  ${}^n\tilde{\mathbf{p}}'$  is  $N^{-1}MN$ . That is, the effect of a change of coordinates on

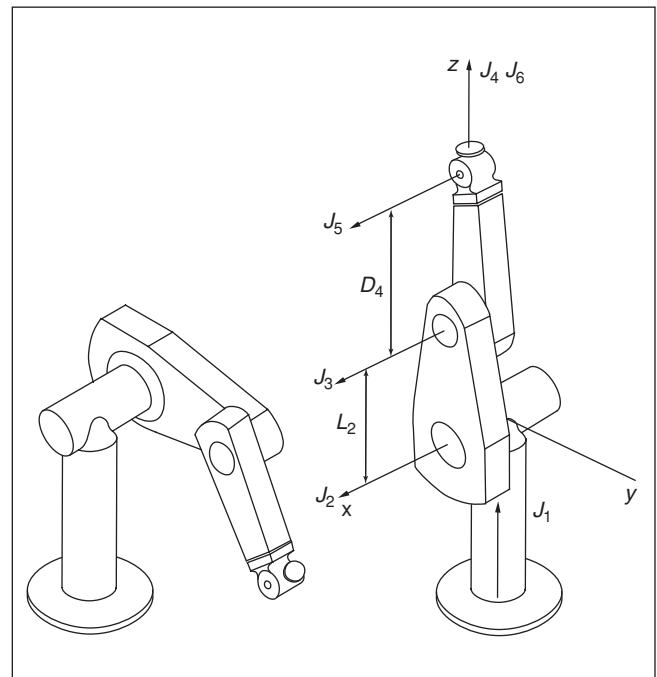


Figure 3. The PUMA robot and its home configuration.

a transformation matrix is a conjugation. Or with the extra layer of notation,  ${}^uM = N^{-1} {}^fMN$ . Notice that  $N$  is expressed in the original fixed coordinate frame.

## Forward Kinematics of Serial Manipulators

In this section the forward kinematics of a simple six-revolute (6-R) manipulator will be derived using the active viewpoint and without using the traditional Denavit-Hartenberg method. The 6-R manipulator is used here for simplicity; the method extends easily to robots with prismatic and even helical joints.

The forward kinematics of a robot relates the displacement of the joints (joint angles for revolute joints) to the position and orientation of the tool or end effector of the robot. The inverse kinematics is the inverse problem of finding the set of joint angles, which will place the end effector of the robot at a specified position.

Instead of using the Denavit-Hartenberg parameters of the robot, we use the home position of the joint axes. For each joint axis, we can find the  $A$  matrix representing rotations around that joint. Often, in the home position of the robot, the joints will be aligned with the coordinate axes so the rotation matrix part of the transformation will be easy to write down. To find the translation part, we need to know the coordinates of a single point on the axis, then we can use the results of the previous section to find the full  $4 \times 4$  transformation matrix. We will look at a specific example later.

To find the overall transformation undergone by the robot's end effector, we simply multiply the  $A$  matrices. This can be justified as follows: Begin with the last joint, the one furthest from the base of the robot (sometimes called the distal joint); a rotation around this joint will not affect the positions of any of the joints nearer the base. Next, we can rotate around the next-to-last joint, again without affecting any joint nearer the base. Continuing in this sequence, we end by rotating around the first joint (the proximal joint). Suppose that  $A_i(\theta_i)$  is the matrix representing a rotation around the  $i$ th joint by an angle of  $\theta_i$ , then the above sequence of transformations can be written as

$$K(\theta_1, \dots, \theta_6) = A_1(\theta_1)A_2(\theta_2)A_3(\theta_3)A_4(\theta_4)A_5(\theta_5)A_6(\theta_6).$$

This compound transformation represents the motion undergone by the robot's end effector from its home position to the position when the joint angles are set to  $\theta_1, \theta_2, \dots, \theta_6$ . Notice that when the joint angles are all zero, the robot will be in its home position.

As a concrete example, we will find the  $A$  matrices for a PUMA robot (see Figure 3). The first thing we need to do is to fix a convenient coordinate frame and home configuration for the robot. In this case, since the first two joints meet at right angles, it is sensible to choose the origin of our coordinates at this meeting point and align two of the

coordinate axes with the home positions of Joints 1 and 2. We can draw up a short list of the directions of the joints in the home configuration and convenient points on the joint axes:

Joint	$\mathbf{v}$	$\mathbf{p}$
$J_1$	$\mathbf{k}$	$\mathbf{0}$
$J_2$	$\mathbf{i}$	$\mathbf{0}$
$J_3$	$\mathbf{i}$	$L_2\mathbf{k}$
$J_4$	$\mathbf{k}$	$D_3\mathbf{i}$
$J_5$	$\mathbf{i}$	$(L_2 + D_4)\mathbf{k}$
$J_6$	$\mathbf{k}$	$D_3\mathbf{i}$

Here,  $\mathbf{i}$ ,  $\mathbf{j}$ , and  $\mathbf{k}$  are unit vectors in the  $x$ ,  $y$ , and  $z$  directions, respectively. The fixed lengths  $L_2$ ,  $D_3$ , and  $D_4$  are the design parameters of the robot.

Now it is a simple matter to find the  $A$  matrices. The first one is simply a rotation around the  $z$ -axis:

$$A_1(\theta_1) = \begin{bmatrix} \cos \theta_1 & -\sin \theta_1 & 0 & 0 \\ \sin \theta_1 & \cos \theta_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

The second one is also simple, this time a rotation around the  $x$ -axis:

$$A_2(\theta_2) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \theta_2 & -\sin \theta_2 & 0 \\ 0 & \sin \theta_2 & \cos \theta_2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

For  $A_3$ , we need to know the term  $(I - R)\mathbf{p}$ . The rotation is again around the  $x$ -axis:

$$\left\{ \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} - \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta_3 & -\sin \theta_3 \\ 0 & \sin \theta_3 & \cos \theta_3 \end{bmatrix} \right\} \times \begin{bmatrix} 0 \\ 0 \\ L_2 \end{bmatrix} = \begin{bmatrix} 0 \\ L_2 \sin \theta_3 \\ L_2(1 - \cos \theta_3) \end{bmatrix},$$

so that,

$$A_3(\theta_3) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \theta_3 & -\sin \theta_3 & L_2 \sin \theta_3 \\ 0 & \sin \theta_3 & \cos \theta_3 & L_2(1 - \cos \theta_3) \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Similarly, we have

$$A_4(\theta_4) = \begin{bmatrix} \cos \theta_4 & -\sin \theta_4 & 0 & (1 - \cos \theta_4)D_3 \\ \sin \theta_4 & \cos \theta_4 & 0 & -\sin \theta_4 D_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

and also

$$A_5(\theta_5) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \theta_5 & -\sin \theta_5 & (L_2 + D_4) \sin \theta_5 \\ 0 & \sin \theta_5 & \cos \theta_5 & (L_2 + D_4)(1 - \cos \theta_5) \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Finally, we have

$$A_6(\theta_6) = \begin{bmatrix} \cos \theta_6 & -\sin \theta_6 & 0 & (1 - \cos \theta_6)D_3 \\ \sin \theta_6 & \cos \theta_6 & 0 & -\sin \theta_6 D_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

This is really all there is to the forward kinematics of serial robots.

However, it is often useful to express things in the tool frame of the robot. The tool frame is a coordinate frame fixed in the tool or end effector of the robot (see Figure 4). A common way to express the forward kinematics of a robot is to give the transformation that takes the standard fixed frame, now called the *world frame*, to the tool frame. This is now easily done; remember that  $K = K(\theta_1, \dots, \theta_6)$  is the transformation that takes the end effector from its home position to its current position, so it will also transform the tool frame from its home position to its current position. So, all we need to do is to first transform from the world frame to the home position of the tool frame. This transformation doesn't depend on the joint angles, so let's just label it  $B$ . In this style, the forward kinematics can be expressed as

$${}^0K = A_1(\theta_1)A_2(\theta_2)A_3(\theta_3)A_4(\theta_4)A_5(\theta_5)A_6(\theta_6)B. \quad (1)$$

The notation  ${}^0K$  is intended to signify the transformation from the initial (0th) frame to the final (6th) frame; see the next section for more details of this notation.

In the example above, we could choose the origin of the tool frame for the PUMA to be located at the center of the wrist. The transformation  $B$  would then be a pure translation:

$$B = \begin{bmatrix} 1 & 0 & 0 & D_3 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & L_2 + D_4 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

As suggested above, a more useful application of the tool frame is to specify moves in terms of the tool frame. For example, suppose that the  $z$ -axis of the tool frame is aligned with an axis in the robot's gripper; now, we might want to command the robot to move a little way along this axis or perhaps turn around this axis. Suppose the desired motion is given by the transformation  $K_\Delta$  expressed in the tool frame. In order to express this in the world frame, we simply conjugate by the appropriate transformation; this is the transformation that takes the tool frame to the world frame, i.e.,  $({}^0K)^{-1}$ . Hence, in the world frame, the move is

$${}^0K_\Delta = {}^0K K_\Delta ({}^0K)^{-1}.$$

One textbook that gives more details of the active view of kinematics is [6]. Unfortunately, this book is now out of print.

## Denavit-Hartenberg Revisited

In this section, the standard approach to the forward kinematics of serial robots will be reviewed. Any standard textbook on robotics will give a fuller account of this material. See, for example, [1]–[5] and [7]. The aim here, however, is to relate the standard method to the approach outlined previously.

The first step here is to attach a coordinate frame to each link of the robot, including the base link. There are detailed rules specifying how to do this, but in essence, the frame in the  $i$ th link is placed so that its  $z$ -axis is aligned with the axis of the  $(i + 1)$ th joint. The first frame is placed in the base link (link 0) and is the world frame. The frame in the last link (link 6) is the tool frame.

The idea is to give the transformation from the world frame to the tool frame. The transformation from frame  $i$  to frame  $j$  is usually expressed as  ${}^i_jM$ , so that the coordinates of a point  $\tilde{\mathbf{p}}$ , expressed in the two frames satisfies

$${}^i\tilde{\mathbf{p}} = {}^i_jM {}^j\tilde{\mathbf{p}}.$$

So we can think of  ${}^i_jM$  as the active transformation that moves frame  $i$  to frame  $j$ .

The overall transformation from the world frame to the tool frame can be found by multiplying the transformation

$${}^0K = {}^0_1M {}^1_2M {}^2_3M {}^3_4M {}^4_5M {}^5_6M.$$

The order of multiplication is, of course, very important. For active transformations, the next transformation in a sequence would multiply on the left. Here, the next

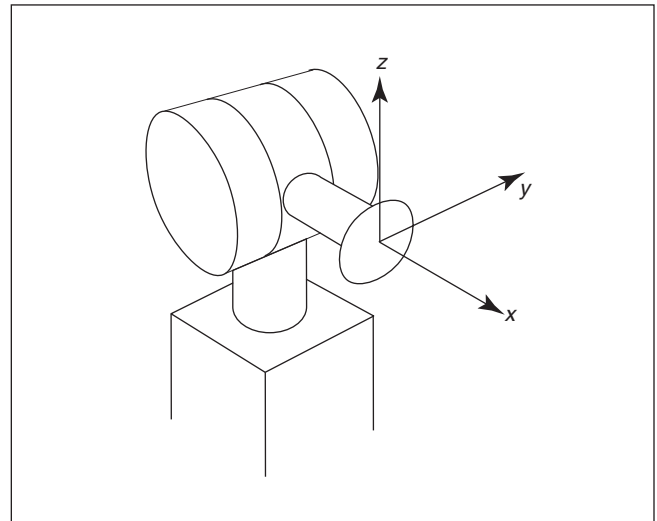


Figure 4. The tool frame attached to the last link of a robot.

transformation multiplies on the right. This is because each  ${}^i_jM$  is expressed in frame  $i$ . To express  ${}^j_kM$  in frame  $i$ , we would conjugate by  ${}^i_jM^{-1}$  to get  ${}^i_jM^{-1}{}^j_kM$ . Now that both transformations are expressed in frame  $i$ , we can multiply on the left, and we get

$${}^i_jM^{-1}{}^j_kM^{-1}{}^i_jM = {}^i_jM^{-1}{}^j_kM.$$

An alternative way to see this is to look at the effect on a point  $\mathbf{p}$  again. Suppose  ${}^i\tilde{\mathbf{p}} = {}^i_jM {}^j\tilde{\mathbf{p}}$  as before, but now we have a third frame  $k$  so that we also have  ${}^j\tilde{\mathbf{p}} = {}^j_kM {}^k\tilde{\mathbf{p}}$ . Substituting for  ${}^j\tilde{\mathbf{p}}$  gives  ${}^i\tilde{\mathbf{p}} = {}^i_jM {}^j_kM {}^k\tilde{\mathbf{p}}$ .

Now each transformation matrix  ${}^i_jM$  has the following structure:

$${}^i_jM = R_z(\theta_j) {}^i_jB.$$

Here, the matrix  ${}^i_jB$  is the transformation from frame  $i$  to frame  $j$  when the robot is in its home position, i.e., when the joint angles are zero. The matrix  $R_z(\theta_j)$  gives the rotation around the  $j$ th joint axis. Remember that in this formalism, every joint axis is aligned with the local  $z$ -axis. It is only the  ${}^i_jB$  transformations that involve the Denavit-Hartenberg parameters, the design parameters of the robot.

Finally we can show that the above is equivalent to (1). If we conjugate the  $A$  matrices above to put them into local coordinates, we get

$$\begin{aligned} R_z(\theta_1) &= A_1(\theta_1) \\ R_z(\theta_2) &= ({}^0_1B)^{-1} A_2(\theta_2) ({}^0_1B) \\ R_z(\theta_3) &= ({}^0_1B {}^1_2B)^{-1} A_3(\theta_3) ({}^0_1B {}^1_2B) \\ &\vdots \\ R_z(\theta_6) &= ({}^0_1B {}^1_2B \cdots {}^5_6B)^{-1} A_6(\theta_6) ({}^0_1B {}^1_2B \cdots {}^5_6B). \end{aligned}$$

We also need the overall transformation from the world frame to the tool frame in the robot's home position; this is clearly

$$B = {}^0_1B {}^1_2B {}^2_3B {}^3_4B {}^4_5B {}^5_6B.$$

Now the two versions of the forward kinematics are exactly equivalent.

## Concluding Remarks

The aim of this article has been to show that the active approach is simple to learn and teach. Students (and teachers) often get confused by having many coordinate frames. An extra level of notation needs to be introduced to distinguish between objects that are the same but expressed in different coordinate frames. When performing multiplication of transformations, we need to make sure that the multiplication is well defined. For active

transformations, this simply amounts to ensuring that they are all expressed in the same coordinate frame, which is trivial if there is only one frame, and getting the order of multiplication correct. For passive transformations, there are more considerations, certainly the multiplication will be well defined if the transformations are expressed in a common frame. However, as we saw above, it is sometimes possible to multiply transformations expressed in different frames.

Mathematically, the active and passive views are equivalent. There may be practical differences between the two approaches. However, it would seem likely that some ideas are better expressed in one format and others are most easily dealt with in the other. So, there is really nothing to choose between them. In the end, the choice is a matter of personal preference and depends on familiarity and personal taste.

## Keywords

Rigid transformations, kinematics.

## References

- [1] J.J. Craig, *Introduction to Robotics: Mechanics and Control*, 2nd ed. Reading, MA: Addison-Wesley, 1986.
- [2] J.K. Davidson and K.H. Hunt, *Robots and Screw Theory: Applications of Kinematics and Statics to Robotics*. London, UK.: Oxford Univ. Press, 2004.
- [3] K.S. Fu, R.C. Gonzalez, and C.S.G. Lee, *Robotics—Control, Sensing, Vision and Intelligence*. New York: McGraw-Hill, 1987.
- [4] P.J. McKerrow, *Introduction to Robotics*. Reading, MA: Addison Wesley, 1990.
- [5] R.P. Paul, *Robot Manipulators: Mathematics, Programming, and Control*. Cambridge, MA: MIT Press, 1981.
- [6] J.M. Selig, *Introductory Robotics*. Englewood Cliffs, NJ: Prentice Hall, 1992.
- [7] M.W. Spong and M. Vidyasagar, *Robot Dynamics and Control*. New York: Wiley, 1989.

**J.M. Selig** graduated from the University of York, U.K., in 1980 with a B.Sc. in physics. He went on to study in the Department of Applied Maths and Theoretical Physics at the University of Liverpool and was awarded a Ph.D. in 1984. From 1984–1987, he was a postdoctoral fellow in the design discipline of the Open University, studying robot gripping. He joined the Department of Electrical and Electronic Engineering at South Bank Polytechnic in 1987. In 1999, he transferred to the School of Computing, Information Systems, and Mathematics. His research interests can be summarized as the applications of modern geometry to problems in robotics.

**Address for Correspondence:** J.M. Selig, Faculty of Business, Computing and Information Management, London South Bank University, London SE1 0AA, UK. Phone: +44 20 7815 7461. E-mail: seligjm@lsbu.ac.uk.